Quadratic Classification

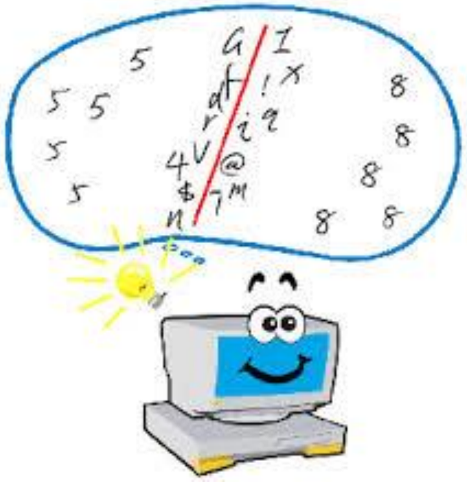
Reporter : 王士銘

Outline

- Classification
- Minimum Error – Maximum A Posterior
- Estimation of parameters
  - Maximum Likelihood
  - Bayesian Estimation
- Discernment Function
- Conclusion

Classification

- Given an object, tell which class it belongs to
- Step -
  - Learn from training objects with class labels
  - Classifying testing objects without class labels
- Assumption - Each class of object follow a probability distribution with some parameters
- Machine learning theory guarantees small training error leads to small testing error
- The object of classifying algorithm is thus minimizing training error



Notations

- c – number of classes
- D – training data
- D_i - subset of training data with elements belonging to class i
- N, n_i – size of D and D_i
- $\boldsymbol{x}, y$ – feature vector and class label representing a training object
- $\hat{y}$ - underground class label
- θ_i – parameters of the probability distribution of class i
- $\widehat{\theta_i}$ - maximum likelihood estimator for θ_i

Minimum Error (Testing)

- Consider 2 classes. Given training data D, to classify a testing object **x**
- $P(error|x) = \begin{cases} 1 - P(y_1|x,D) \text{ , } if\ we\ choose\ y = 1 \\ 1 - P(y_2|x,D) \text{ , } if\ we\ choose\ y = 2 \end{cases}$
- Hence, choose class 1 if $P(y_1|\boldsymbol{x}, D) > P(y_2|\boldsymbol{x}, D)$
- Minimizing error rate is equivalent to maximizing $P(y|\boldsymbol{x}, D)$, posterior probability
- $P(y_i|\boldsymbol{x}, D) = \frac{P(\boldsymbol{x}|y_i, D\ )P(\boldsymbol{y}_i|D)}{P(\boldsymbol{x},D)} = \frac{P(\boldsymbol{x}|y_i, D_i\ )P(y_i)}{P(\boldsymbol{x},D)}$
- Independence : $D_j, j \neq i$ gives no more information to $P(x|y_i, D_i)$
- Independence : D, y_i are independent
- We need to estimate$P(\boldsymbol{x}|y_i, D)$and $P(y_i)$ for $i = 1{,}2 \ldots, c$

Estimation of Parameters(Training)

- need to estimate$P(\boldsymbol{x}|y_i, D)$and $P(y_i)$ for $i = 1,2 \dots, c$
- $P(y_i) = \frac{n_i}{N}$
- $P(\boldsymbol{x}|y_i, D_i) = \int P(x|\theta_i, D_i)P(\theta_i|D_i)d\theta_i = \int P(x|\theta_i)P(\theta_i|D_i)d\theta$
- $P(\theta_i|D_i) = \frac{P(D_i|\theta_i)P(\theta_i)}{P(D_i)}$
- Maximum Likelihood Estimation(MLE) : estimate $\widehat{\theta_i} = \arg_{\theta} max\, \mathrm{P}(\mathrm{D}|\theta_i)$ and let $P(\theta_i|D_i) = \delta_{\widehat{\theta_i}}(\theta)$ => $P(x|y_i, D_i) = P(x|\theta_i)$
- Bayesian estimation : $P(\theta_i|D_i) = \frac{P(D_i|\theta_i)P(\theta_i)}{P(D_i)} = \alpha\{\Pi_{\mathrm{k}=1}^{n_i} P(x_k|\theta_i)\}P(\theta_i)$
- If both $P(\theta_i)$ and $P(x|\theta_i)$ are Gaussian, then$P(\theta_i|D_i)$ is Gaussian

Maximum Likelihood Estimation(Training)

- $P(\theta_i|D_i) = \frac{P(D_i|\theta_i)P(\theta_i)}{P(D_i)} = \alpha\{\Pi_{\mathrm{k}=1}^{n_i} P(x_k|\theta_i)\}P(\theta_i)$
- Estimate $\widehat{\theta_i} = \arg_{\theta} max\ \mathrm{P(D}|\theta_i)$ and let $P(\theta_i|D_i) = \delta_{\widehat{\theta_i}}(\theta)$
- Define log-likelihood function $l(\theta_i) \equiv lnP(D_i|\theta_i) = \Sigma_{k=1}^{n} \ln P(x_k|\theta_i)$
- $\widehat{\theta_i} = \arg_{\theta_i} max\ l(\theta_i)$ =>$\nabla_{\theta_i} l(\widehat{\theta_i})$=0

where $$\nabla_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_d} \end{bmatrix}$$

Maximum Likelihood Estimation(Training)

- Consider normal distribution for each class $P(\boldsymbol{x}|\theta_i) = N(\boldsymbol{x}|\mu_i, \Sigma_\mathrm{i})$
- $l(\theta_i) = \Sigma_{k=1}^{n_i} lnN(x_k|\mu_i, \Sigma_i)$ = $\frac{\mathrm{n_i}}{2} \ln|\Sigma| + \frac{-1}{2} \Sigma_{k=1}^{n_i} (x_k - \mu_i)^T \Sigma_\mathrm{i}^{-1} (x_k - \mu_i) + constant$
- Taking the derivative over μ_i and set it to zero :
- $\Sigma_{k=1}^{n_i} \Sigma_\mathrm{i}^{-1} (x_i - \mu_i) = 0$
- $\widehat{\mu_i} = \frac{1}{n_i} \Sigma_{k=1}^{n_i} x_i$

Maximum Likelihood Estimation (Training)

- $l(\theta_i) = \frac{\mathrm{n_i}}{2} \ln|\Sigma| + \frac{-1}{2} \Sigma_{k=1}^{n_i} (x_k - \mu_i)^T \Sigma_{\mathrm{i}}^{-1} (x_k - \mu_i) + constant$
  $\propto \frac{\mathrm{n_i}}{2} l\,\mathrm{n}|\Sigma| + \frac{-1}{2} \Sigma_{k=1}^{n_i} Trace\left(\Sigma^{-1} (x_k - \mu_i)^T (x_k - \mu_i)\right)$
  $= \frac{\mathrm{n_i}}{2} l\,\mathrm{n}|\Sigma| + \frac{-1}{2} Teace(\Sigma_{k=1}^{n_i} \Sigma^{-1} (x_k - \mu_i)^T (x_k - \mu_i))$
- Taking the derivative over Σ_i and set it to zero using
  1) $\frac{\partial}{\partial \Sigma} \ln|\Sigma| = \Sigma^{-T}$
  2) $\frac{\partial}{\partial \Sigma} Tr(\Sigma A) = \frac{\partial}{\partial \Sigma} Tr(A\Sigma) = \mathrm{A^T}$
- $\hat{\Sigma}_i = \frac{1}{n_i} \Sigma_{k=1}^{n_i} (x_k - \widehat{\mu_i})(x_k - \widehat{\mu_i})^T$

Discriminant Function (Testing)

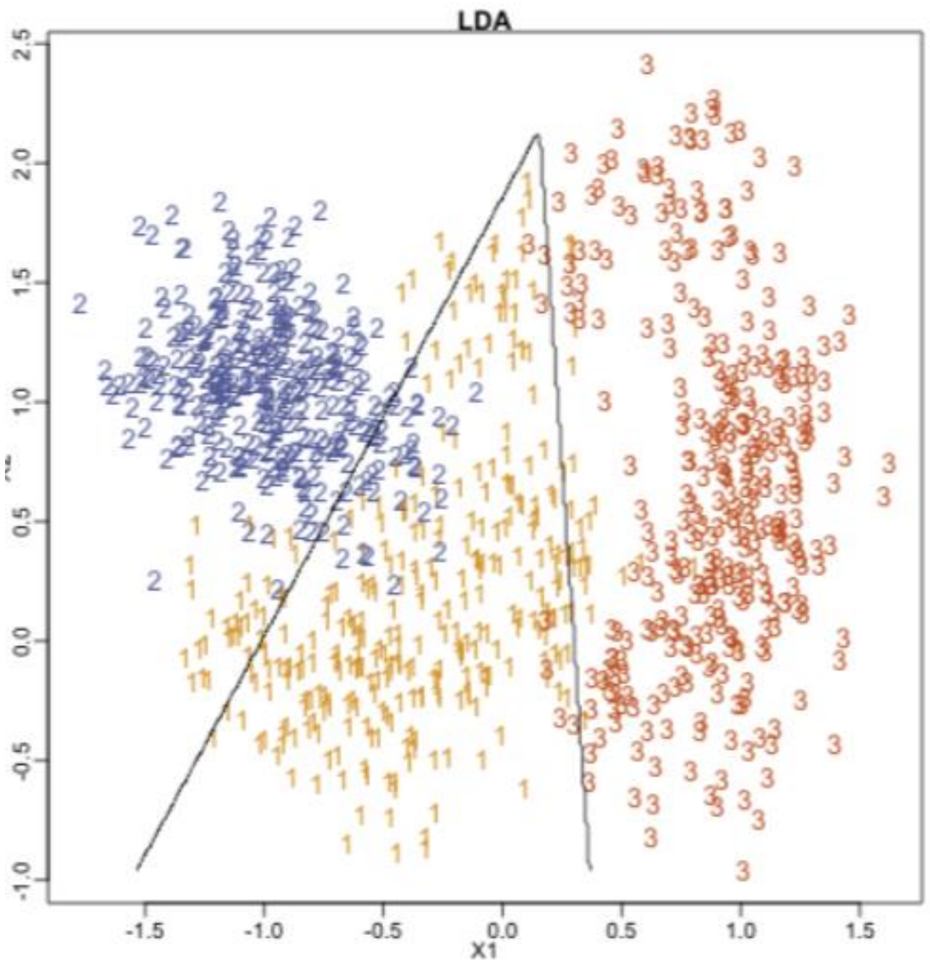
- Suppose we model distribution of each class as multivariate Gaussian and estimate the mean μ_i and covariance matrix Σ_i by MLE
- $P(y_i|\boldsymbol{x}) = \frac{P\left(\boldsymbol{x}\middle|\widehat{\theta_i}\right)P(\boldsymbol{y}_i)}{P(\boldsymbol{x})} = \frac{N(\boldsymbol{x}|\widehat{\mu_i},\widehat{\Sigma_i})P(y_i)}{P(\boldsymbol{x})}$
- Choose class i if $P(y_i|\boldsymbol{x}, D) > P(y_j|\boldsymbol{x}, D)$ for $j \neq i$
- Define discrimination function $\delta_{ij}(\boldsymbol{x}) = \ln(P(y_i|\boldsymbol{x}, D)/P\left(y_j\middle|\boldsymbol{x}, D\right))$
- Choose i instead of j if $\delta_{ij}(x)$>0

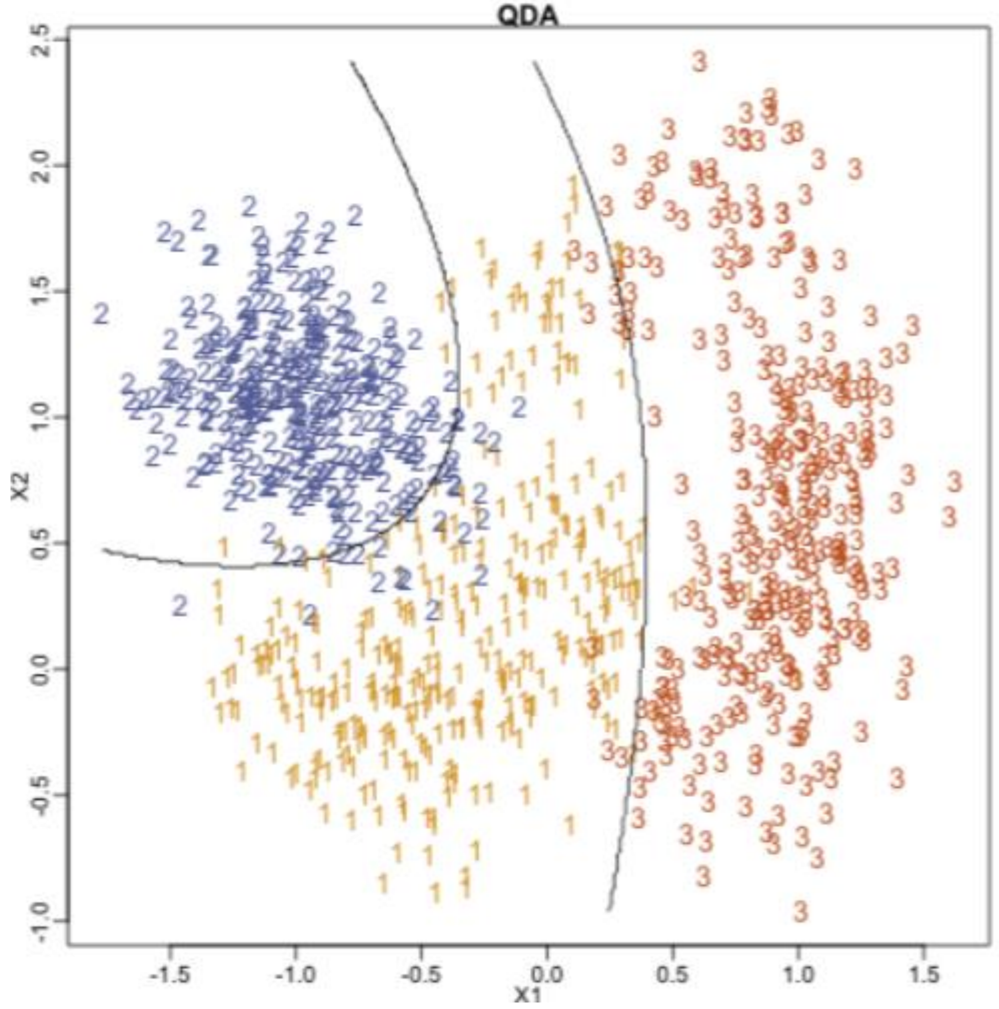
Discriminant Function (Testing)

- Discrimination function $\delta_{ij}(\boldsymbol{x}) = \ln(P(y_i|\boldsymbol{x}, D)/P(y_j|\boldsymbol{x}, D))$ determines the boundary for class i and j
- If we assume $\widehat{\Sigma_\mathrm{i}} = \Sigma$ for $i = 1,2, \ldots, c$
- $\delta_{ij}(\boldsymbol{x}) = (\widehat{\mu}_i - \widehat{\mu_0})^T\, \widehat{\Sigma}^{-1}\boldsymbol{x} - \frac{1}{2}(\widehat{\mu_j} - \widehat{\mu}_i)^T\, \widehat{\Sigma}^{-1}(\widehat{\mu_j} - \widehat{\mu}_i) + \ln\frac{P(y_i)}{P(y_j)}$

  = $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{w_0}$ **->** linear discernment function
- If we assume different $\widehat{\Sigma_\mathrm{i}} = \Sigma$ for $i = 1,2, \ldots, c$
- $\delta_{ij}(\boldsymbol{x})$ will be quadratic function of $\boldsymbol{x}$ **->** quadratic discernment function

Discriminant Function (Testing)





Conclusion

- A traditional algorithm but has not bad performance
- Chinese handwritten recognition with Gabor filter feature

QDA	0.8724
Neural Network	0.8870
SVM	0.918945